

A SoC Interconnexion Scheduling

Design and Implementation of a Scheduler Based on Credited-iSLIP Algorithm

Dhia BELHAJALI¹, Ihsen BEN MBAREK², Salem HASNAOUI³, Khaled JELASSI⁴

Networking and Distributed Computing, Sys'Com- Sys'Com- Sys'Com- Electrical Systems Laboratory, National Engineering School of Tunis, BP 37, LE BELVEDERE 1002 TUNIS, Tunisia

¹dhia.belhajali@yahoo.fr; ²ben_mbarek_ihsen@yahoo.fr; ³Salem.Hasnaoui@enit.rnu.tn; ⁴jelassi_2000@yahoo.com

Abstract

A SoC (System-on-Chip) interconnection requires crossbar switches to connect inputs to outputs. These devices must be configured by a scheduling algorithm that allows arriving packets to be delivered across the switch efficiently and quickly. A suitable scheduling algorithm has to achieve the maximum throughput while maintaining stability and eliminating starvation.

Thus, in this paper, we present a design and an implementation of a scheduler for a 32x32 SoC interconnexion based on an iterative scheduling algorithm that we called Credited-iSLIP (iterative Serial Line Internet Protocol). It is a variation of the iSLIP algorithm. A description of the scheduler designed is presented with simulation results to indicate its performance.

The design has been implemented in VHDL RTL, simulated with Modelsim 6.2 and synthesized using Xilinx-ISE and the Virtex-4 device XC4VFX100-12FFG1517 of 90 nm technology to achieve a maximum frequency of 265.88 MHz, a minimum slices utilization of 771 and total estimated power consumption about 915 mW. The number of iterations is fixed to $i = 8$.

Keywords

Network-on-Chip; VOQ; Scheduler; Credited-iSLIP Algorithm; VHDL; Virtex-4; Xilinx-ISE

Introduction

Nowadays, with the impressive revolution of electronics, the industry tends to converge several systems to one multifunction device such as PC multimedia and mobile phones. Thus, a new technique has appeared called System-on-Chip (SOC) design in which pre-designed and pre-verified virtual components called Intellectual Properties (IP) blocks or IP cores- are combined on the same chip. These IP blocks can be reused to design new circuits and may include embedded processors, memory blocks, interface blocks and components that handle specific tasks. Thus, SoC designs offer a suitable solution to integrate complex systems on a single chip with the

advanced VLSI technology while ensuring a very high quality of service (QoS) with a low cost.

The increase in the complexity of SoC designs and the tendency to decrease the time-to-market of the products integrating these systems are factors requiring the adoption of appropriate design methodology. The major difficulty in the design of such systems is the communication between different modules inside it. A performing and economic solution is to rely on a predefined configurable communication infrastructure, called Network-on-Chip (NoC). The use of a NoC is required, especially, for large SoC interconnexions for technological (asynchronism between distant parts of the same chip) and functional (reuse of already designed blocks) reasons.

The challenging of such an interconnection is the manner to schedule the transfer of packets from input to output SoCs through a central switching fabric, therefore, the designer must use the appropriate scheduling algorithm for his scheduler that allows multiple packets to be delivered to their outgoing port quickly and efficiently.

In this paper, we present a design and an implementation of a hardware scheduler based on a scheduling algorithm -called "Credited-iSLIP (iterative Serial Line Internet Protocol)"- that configures a VOQ (Virtual Output Queues) crossbar switch for a 32x32 SoC interconnexion. This algorithm attempts to achieve a maximum throughput, to eliminate starvation and to be fast and simple to implement in hardware. A VOQ crossbar switch is used in our interconnexion because it is simple to implement and is non-blocking.

VOQ Crossbar Switch

The crossbar is one of the simplest and the most used architectures for the design of high performance

switches. In spite of its non-blocking capability and simplicity, a crossbar switch has a critical drawback. Indeed, it, usually, uses input queues to store incoming packets, when a packet arrives to an input it is stocked into a single First In First Out (FIFO) queue. Thus, the centralized scheduler considers only the packet at the head of each FIFO queue, which blocks all packets behind it that need to be delivered to different outputs. This phenomenon, called Head Of Line (HOL) blocking, limits the throughput to 58,6% [9]. The HOL effect is eliminated by the employment of Virtual Output Queues (VOQ) [6]. In fact, instead of using a single FIFO for all packets, each input maintains a separate queue for each output as shown in Fig.1.

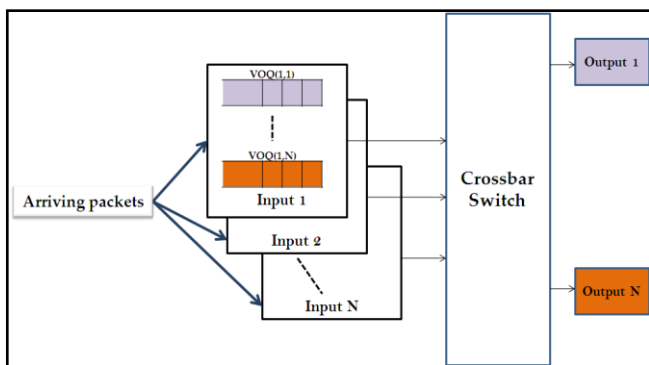


FIG.1 A VOQ CROSSBAR SWITCH

When VOQs are used, the throughput of crossbar switch increases from 58.6% to 100%.

The Credited-iSLIP algorithm we used, configures the crossbar switch to determine how N^2 VOQs will be served at a given time slot. To understand its operation, we start by describing iSLIP algorithm then in the second section, we depict Credited-iSLIP algorithm and finally we present the scheduler design with simulation results.

The iSLIP Algorithm

The iSLIP algorithm is an iterative, Round-Robin algorithm. Actually, it is based on Parallel Iterative Matching (PIM) and Round-Robin Matching algorithms.

Parallel Iterative Matching

The PIM algorithm attempts to quickly find a conflict-free match in several iterations, each of which consists of three steps as shown in Fig. 2:

- Request: each unmatched input notifies outputs for which it has a queued cell by sending a request, noting that one output can

be requested by many inputs.

- Grant: each unmatched output that receives any request, chooses randomly only one to grant, noting that one input can be granted by many outputs (if it made many requests).
- Accept: if an input receives any grant, it accepts randomly one and notifies the output chosen.

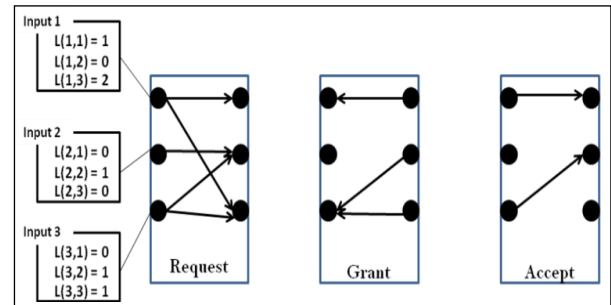


FIG.2 EXAMPLE OF ONE ITERATION OF THE PIM ALGORITHM FOR 3X3 SWITCH, NOTE $L(I,J)$ = SIZE OF THE VOQ(I,J)

At the end of an iteration, only unmatched inputs and outputs can be connected in the next.

The use of randomness allows the PIM algorithm to avoid starvation and to reduce the number of iterations that it takes to complete.

For an $N \times N$ switch, the algorithm converges after $O(\log N)$ iterations, in fact, after each iteration, the number of remaining possible connections is reduced by an average of at least $\frac{2}{3}$.

Although the use of random arbiters will often permit the algorithm to converge quickly, the implementation at high speed is very difficult and expensive. Further, when the algorithm takes only one iteration to complete (when each output grants to a different input), the throughput is limited to about 63%. Finally, under heavy loads, PIM can lead to unfairness between connections.

Round-Robin Matching Algorithm

The RRM algorithm overcomes drawbacks of PIM (complexity and unfairness) with the use of priority arbiters instead of random ones. It consists of three steps:

- Request: the same as PIM algorithm.
- Grant: when an output receives requests, it grants to only one input. Using a fixed round-robin schedule, the granted input is the next one to the highest priority element. The output notifies each input to confirm or not the grant. The grant pointer (gi) points to the highest

priority element and must be incremented (modulo N) to one position beyond the granted one.

- Accept: when an input gets any grant, it accepts only one output. Starting from the highest priority element, the accepted one is that appears next in a fixed round-robin schedule. The highest priority element is pointed by the accept-pointer (a_i) of the round-robin schedule. The latter pointer must be incremented (modulo N) to one position beyond the accepted one.

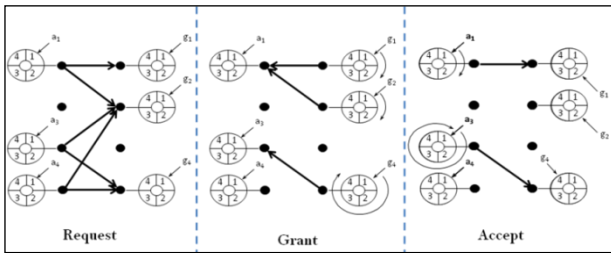


Fig.3 Example Of One Iteration Of The Rrm Algorithm For 4x4 Switch

Although the rotating priority used in round-robin arbiters makes the algorithm fair and simple to be implemented, RRM does not perform well because of the synchronization of grant arbiters resulting from the rules to update the pointers at the output arbiters, which leads to a maximum throughput of just 50%.

Thus, the iSLIP algorithm improves the performance of RRM algorithm by reducing the phenomenon of arbiters' synchronization. In fact, iSLIP is similar to RRM except for a condition on the rules of grant pointers updating. The grant pointer is updated if, and only if, the grant is accepted in the third step. Then, the most recently connected input and output will have the lowest priority in the next iteration. Fig.4 presents an example of one iteration of the iSLIP algorithm.

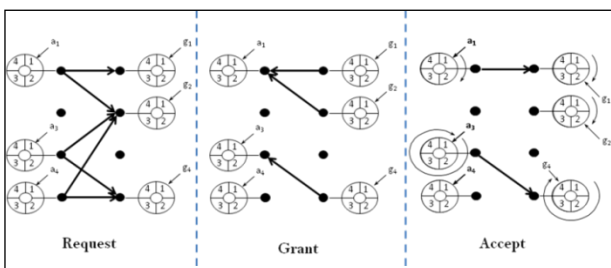


FIG.4 EXAMPLE OF ONE ITERATION OF THE RRM ALGORITHM FOR 4X4 SWITCH

The iSLIP can achieve 100% throughput for uniform traffic which is fair and starvation-free and can be easily implemented in hardware.

The Credited-iSLIP

In our interconnexion, we used a flow control based on a simple credit-debit system. In fact, each output SoC has a counter (credit) initialized to the number of available places in the output buffer. The counter is decremented when the SoC sends a packet and incremented when the SoC receives a CreditIn. The credit informs the input port the number of packets that the output port can receive.

The iSLIP is modified so that outputs with 0 credits must not participate in the current arbitration cycle. In fact, we added to the scheduler a credit signal indicating whether the output has enough downstream credit to send an other transfer. When this signal is asserted, the output is disabled from the grant arbitration.

This algorithm not only shuns starvation and HOL blocking problems, but also reduces burstiness and latency. The Credited-iSLIP converges in $O(\log N)$ where the i-SLIP converges in N iteration. Moreover, throughput under Credited-iSLIP scheduling reaches 100%.

The Scheduler Design

The aim of this design is to manage the transfer of packets across a VOQ crossbar switch providing, then, a fast and efficient interconnexion between 32 SoCs. The scheduler analyzes the VOQs of each input and configures the matching of inputs and outputs, it is based on the Credited-iSLIP algorithm.

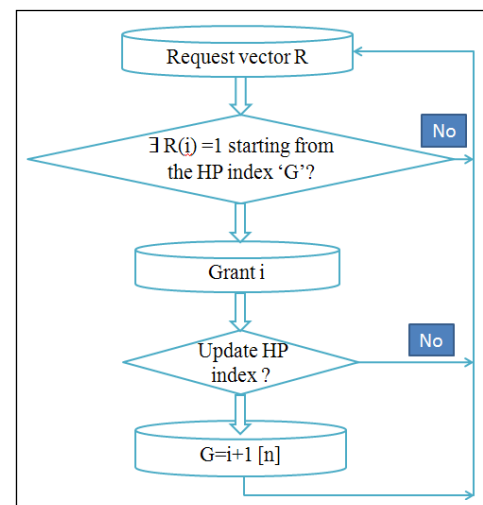


FIG.5 CREDITED-ISLIP ARBITER FLOWCHART

Thus, the scheduler is composed of 32 grant arbiters and 32 accept arbiters. In addition, we need three

input-to-output Swizzles to reorganize request vectors, grant vectors and accept vectors, and a Finite State Machine (FSM) to control the iteration number and arbiters updating.

Both of grant arbiter and accept arbiter have the same architecture. Each arbiter is based on a simple round-robin arbiter for which we added a signal to indicate the rules of its updating priority that we explained above. Fig.5 shows the flowchart of the Credited-iSLIP arbiter.

Actually, each arbiter is composed of programmable priority encoders (PPE) and a state pointer to record which input should have the highest priority on the next arbitration cycle. Fig.6 shows the corresponding flowchart.

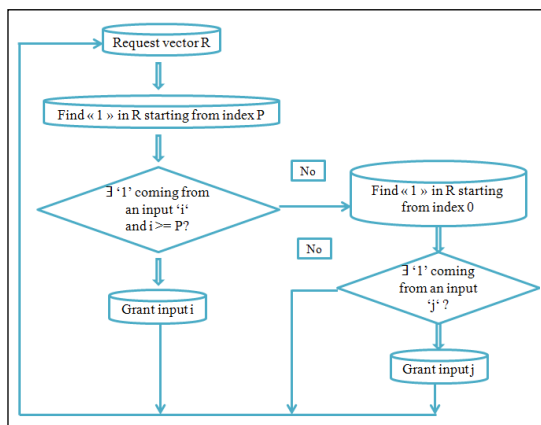


FIG.6 PPE FLOWCHART

In fact, the PPE is the most timing-critical component in the scheduler. Its speed decides how fast the arbitration logic can run. Our PPE combines two simple PEs and adds priority functionality to the PE to allow the arbiter pointer to point to the next request after the current arbitration. Fig.7 shows the flowchart of a simple PE.

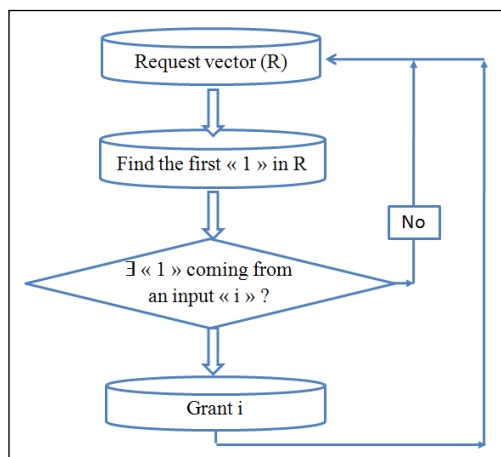


FIG.7 SIMPLE PE FLOWCHART

This PPE design produces minimized delay, and has a smaller area than more classic fast PPE designs.

The Scheduler Simulation

In this section, simulation and implementation results of our scheduler have been presented using Modelsim SE 6.2 full version as VHDL simulator and Xilinx ISE 8.1i WebPACK version as VHDL synthesizer.

We choose to implement our design in VHDL-RTL (Virtual Hardware Description Language-Register Transfer Level) which is an IEEE standard and supported by all major tools vendors. In fact, VHDL is a simulatable language allowing designers to write testbenches which makes the simulation much more portable. In addition, this language is hierarchical, that is it can be used to describe not only the device but entire systems, it helps to clarify the specification at the early stage of the design before any implementation decision. With early verification, problems can be corrected at earlier stage which reduces the overall development time.

The simulated example is characterized with:

- Number of SoCs: $N=32$;
- Number of iterations: $i=8$;
- Packet width = 72bits : Source=3bits, Destination: 3bits and Data=66bits;
- 32 Grant arbiters (output side);
- 32 accept arbiters (input side);
- 3 swizzles;
- 1 Finite State Machine;
- 1 update_enable: a signal included in the arbiter and allows the algorithm to update the priority only under the specific circumstances mentioned above.

The Simulation

The simulation results of our Credited-iSLIP scheduler are illustrated in Fig.8.

Actually, the scheduler attempts to match input ports to output ports. The input ports send an occupancy vector indicating for which outputs they have pending packets. Over 8 iterations, the scheduler matches inputs to outputs. At the end of 8 iterations, the solution is sent to inputs to indicate which packet they should send, and to outputs to select from which input receive the packet. The input signal "output-available" indicates that the output port has enough credit available to receive data.

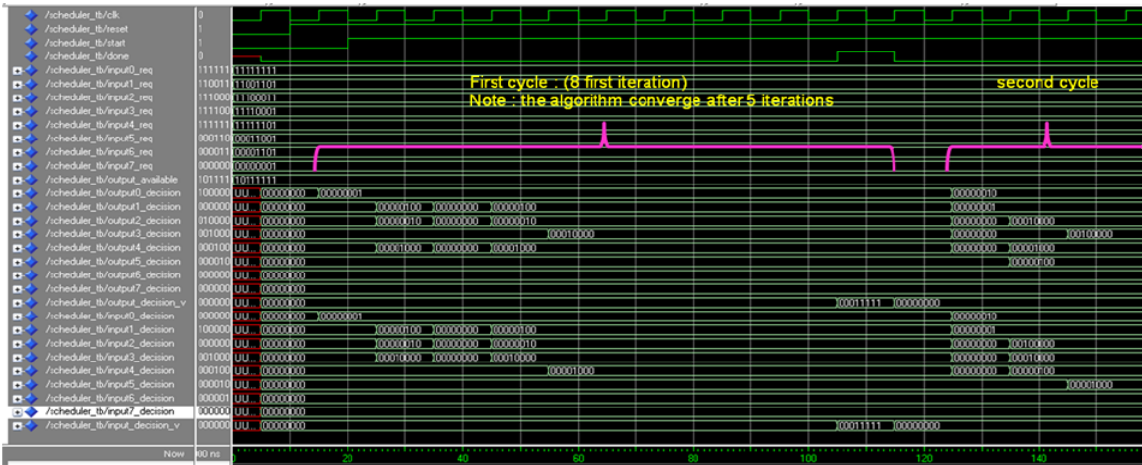


FIG.8 CREDITED-ISLIP SCHEDULER SIMULATION

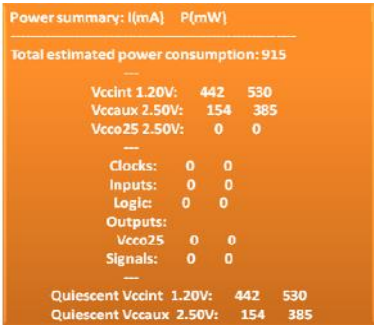


FIG.9 POWER SUMMARY

Synthesis Results

The design was implemented in VHDL RTL, simulated with Modelsim 6.2 and synthesized using Xilinx-ISE and the Virtex-4 device XC4VFX100-12FFG1517 of 90 nm technologies to achieve a maximum frequency of 265.88 MHz, a minimum slices utilization of 771 and a total estimated power consumption about 915 mW (Fig.9).

RTL schematic of the design is presented in Fig.10.

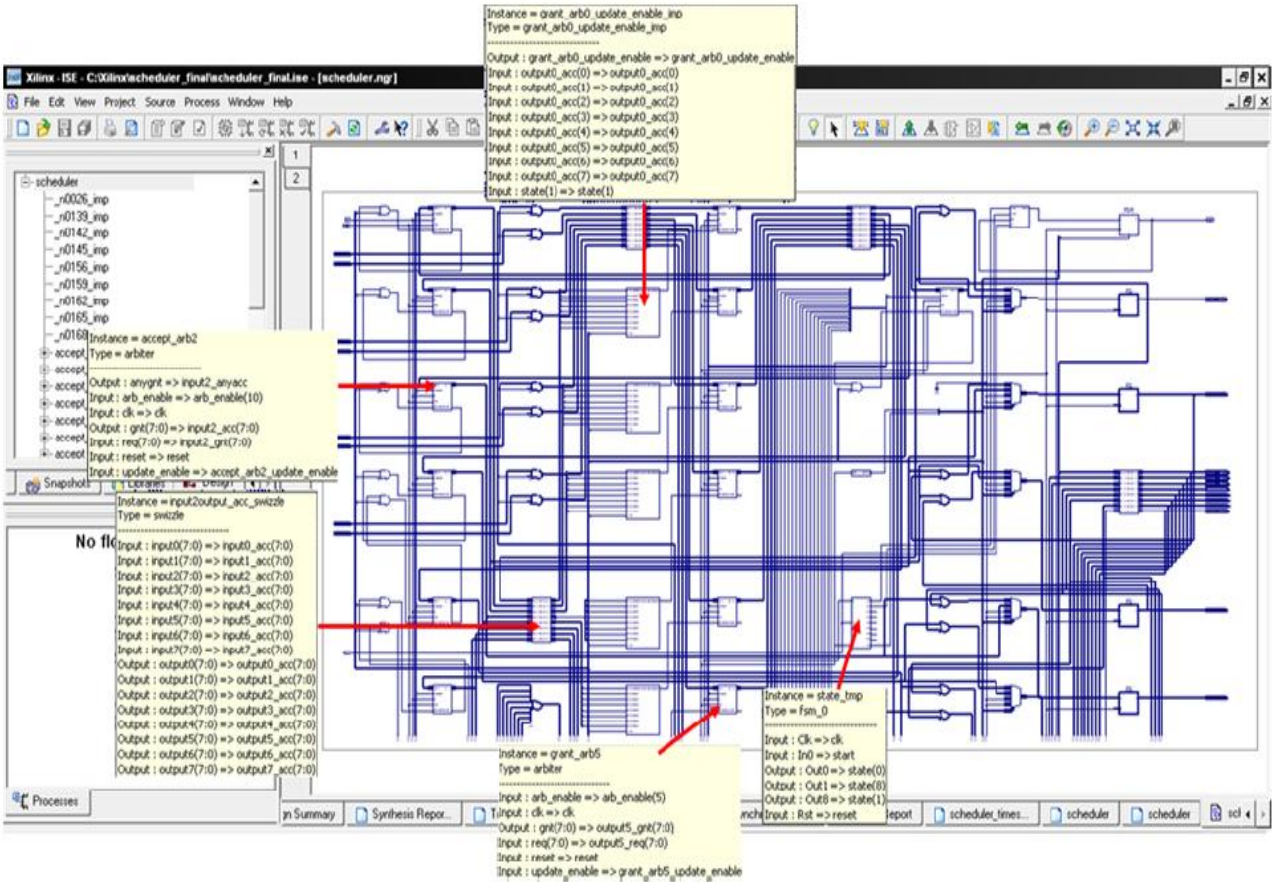


FIG.10 SCHEDULER RTL SCHEMATIC

Conclusions

In this paper we have designed and implemented a scheduler for a 32x32 SoC interconnexion. Since our interconnexion is based on a crossbar switch that uses virtual output queuing, a fast and fair scheduling algorithm is needed, which must be starvation-free and simple to be implemented in hardware as well. Thus, we employed a modified version of the iSLIP algorithm called Credited-iSLIP algorithm. This algorithm attempts to converge in several iterations, to operate at high speed and to achieve 100% throughput for uniform traffic.

REFERENCES

- Anderson, Thomas E., Owicki, Susan S., Saxe, James B., and Thacker, Charles P., "High Speed Switch Scheduling for Local Area Networks", *ACM Trans. Comput. Syst.*, vol. 11, no. 4, pp. 319–352, Nov. 1993.
- Ashenden, Peter J., "VHDL Tutorial", Elsevier Science (USA), 2004.
- Banovic, Dusan and Radusinovic, Igor, "Scheduling algorithm for VOQswitches", *International journal of electronics and communications, (AEÜ)* 62 (2008) 455-458.
- Benini, L. and DeMicheli, G., "Networks on Chips: A New SoC Paradigm," *Computer*, vol. 35, no. 1, pp. 70-78, Jan. 2002.
- Fewer, Colm, « Cross Bar Switch Implemented in FPGAs », Xilinx WP166 (v1.0) September 9, 2002.
- Gospodinov, Mitko and Gospodinova, Evgeniya, "Analysis of iSLIP scheduling algorithm for input-queuing switches", *International Conference on Computer Systems and Technologies - CompSysTech'2004*.
- Gupta, P. and McKeown, N., "Designing and Implementing a Fast Crossbar Scheduler," *IEEE Micro*, 1999, pp. 20-28.
- Ho, Ron, Mai, Kenneth W. and Horowitz, Mark A., "The Future of wires", *Proceedings of IEEE*, vol. 89, no. 4, pp. 490-504, Apr. 2001.
- JOU, JER-MIN AND LEE, YUN-LUNG, "An Optimal Round-Robin Arbiter Design for NoC", *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING* 26, 2047-2058 (2010).
- Kamath, Udaya, Kaundin, Rajita, "System-on-Chip Designs: Strategy for Success", white paper, June 2001.
- Katevenis, Manolis, Passas, Georgios, Simos, Dimitrios, Papaefstathiou, Ioannis and Chrysos, Nikos, « Variable Packet Size Buffered Crossbar (CICQ) Switches", *ICS-FORTH*, SEP. 2003.
- Khan, Mohammad Ayoub and Ansari, Abdul Quaiyum, "Modeling and simulation of 128-bit crossbar switch for Network-on-Chip", *International Journal of VLSI design & Communication Systems (VLSICS)*, vol.2, no.3, September 2011.
- Lee, Tony T. and Liew, Soung C., "Principles of Broadband Switching and Networking" John Wiley & Sons, 2010, pp.115-119.
- McKeown, N., "Scheduling Cells in an Input-Queued Switch", PhD thesis, University of California at Berkeley, May 1995.
- McKeown, Nick, "The iSLIP Scheduling Algorithm for Input-Queued Switches », *IEEE/ACM TRANSACTIONS ON NETWORKING*, VOL. 7, NO. 2, APRIL 1999.
- McKeown, Nick, Anantharam, Venkat and Walrand, Jean, "Achieving 100% Throughput in an Input-Queued Switch", in *Proc. IEEE INFOCOM '96*, San Francisco, CA, pp. 296–302.
- M., Karol, M, Hluchyj and S, Morgan, "Input versus output queueing on a space division switch," *IEEE Trans. Communications*, vol. 35, no. 12, pp.1347-1356, Dec. 1987.
- Pande, Partha Pratim, Cristian Grecu, Michael Jones, Andre Andre' Ivanov and Resve Saleh, "Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures", *IEEE Transactions on computers*, vol. 54, no. 8, pp. 1025-1040, August 2005.
- Pape, J., "Implementation of an On-chip Interconnect Using the i-SLIP Scheduling Algorithm", *Intermediate Report – Specification and Timeline*, 2006, pp.64-71.
- Saleh, Resve, Wilton, Steve, Mirabbasi, Shahriar, Hu, Alan, Greenstreet, Mark, Lemieux, Guy, Pande, Partha Pratim, Grecu, Cristian and Ivanov, Andre, "System-on-Chip: Reuse and Integration" in the *Proceedings of the IEEE* | Vol. 94, No. 6, June 2006.
- Shin, Eung S., Mooney III, Vincent J. and Riley, George F., "Round-robin Arbiter Design and Generation", *ISSS'02*, October 2–4, 2002, Kyoto, Japan.



Dhia BEL HAJ ALI was born in Nabeul, Tunisia in 1985. She is a transmission engineer at Tunisia Telecom. She received the Engineer diploma degree in telecommunications in 2008 and Master degree in Networking and Distributed Computing in 2010 from National School of Engineering of Tunis.

She has been a PhD student since 2011 in the "Networking and Distributed computing" research group within the Communications Systems Laboratory at the National Engineering School of Tunis. Her research interests are distributed applications and Network on Chip design. She has served some conference committees and journals since 2012.



Ihssen BEN MBAREK was born in Mahdia, Tunisia in 1983. She received the degree in Electronic and Microelectronic engineering in 2007, as well as Master degree in Industrial Systems Engineering in 2009 both from the National Engineering School of Tunis in Tunisia. She has been a PhD student since 2010 in the "Networking and Distributed computing" research group within the Communications Systems Laboratory at the National Engineering School of Tunis. She has held a contractual assistant post at the Higher Computer Science Institute of Tunis in Embedded Systems Department also since 2010. Her research interests are the industrial networks for real-time, distributed applications and Network on Chip design. She has served on some conference committees and journals reviewing processes since 2012.



Salem HASNAOUI is a professor at the Department of Computer and Communication Technologies at the National School of Engineering of Tunis. He received the Engineer diploma degree in electrical and computer engineering

from National School of Engineering of Tunis. He obtained a M.Sc. and third cycle doctorate in electrical engineering, in 1988 and 1993 respectively. The later is extended to a PhD. degree in telecommunications with a specialization in networks and real-time systems, in 2000.

He is author and co-author of more than 40 refereed publications, with a patent and a book. His current research interests include real-time systems, sensor networks, QoS control & networking, adaptive distributed real time middleware and protocols that provide performance assured services in unpredictable environments. He is responsible for the research group "Networking and Distributed computing" within the Communications Systems Laboratory at the National School of Engineering of Tunis.

Dr. Hasnaoui has served many conference committees and journals and also the designated inventor of the Patent "CAN Inter-Orb protocol- CIOP and a Transport Protocol for Data Distribution Service to be used over CAN, TTP, FlexCAN and FlexRay protocols".



Khaled JELASSI was born in 1962, Tunisia. He received the B.S degree in Electrical Engineering from the National Engineering School of Monastir, Manastir Tunisia, in 1987, the Ph.D. degree in Engineering from the Institut National Polytechnique de Toulouse, Toulouse, France in 1991 and the Habilitation degree from the National Engineering School of Tunis, Tunisia in 2003.

He is currently Professor at Tunis University, Tunisia and a Researcher with the Electrical System Laboratory at the National Engineering School of Tunis.

Dr. Jelassi research interest is in the field of supervision and diagnosis of the faults of the electromechanical systems composed of ac drives coupled to gear which is driving the load.